High-Level Synthesis with the Vitis HLS Tool

DSP-HLS (v1.0)

Course Specification

Course Description

This course provides a thorough introduction to the Vitis™ High-Level Synthesis (HLS) tool.

The focus of this course is on:

- Converting C/C++ designs into RTL implementations
- Learning the Vitis HLS tool flow
- Creating I/O interfaces for designs by using the Vitis HLS tool
- Applying different optimization techniques
- Improving throughput, area, latency, and logic by using different HLS pragmas/directives
- Exporting IP that can be used with the Vivado™ IP catalog
- Downloading for in-circuit validation

What's New for 2023.1

- Abstract Parallel Programming Model for HLS module: Added information on how to decide whether to use control-driven or data-driven TLP
- HLS Design Flow – System Integration module: Introduced the new Vitis Unified IDE
- All labs have been updated to the latest software versions

Level – DSP 3

Course Details

- 2 days ILT or 16 hours On-Demand
  - 22 lectures
  - 11 labs
  - 4 ILT demos

Price – $1,600 or 16 Xilinx Training Credits

Course Part Number – DSP-HLS

Who Should Attend? – Software and hardware engineers looking to utilize high-level synthesis

Prerequisites

- C or C++ knowledge
- Basic RTL design flow knowledge

Software Tools

- Vitis HLS tool 2023.1
- Vivado Design Suite 2023.1
- Vitis unified software platform 2023.1

Hardware

- Architecture: Zynq™ UltraScale+™ MPSoC and Versal™ AI Core series
- Demo board: Zynq UltraScale+ MPSoC ZCU104 board*

* This course focuses on the Zynq UltraScale+ MPSoC architecture. Check with Morgan Advanced Programmable Systems, Inc. for the specifics of the in-class lab board or other customizations.

After completing this comprehensive training, you will have the necessary skills to:

- Enhance productivity using the Vitis HLS tool
- Describe the high-level synthesis flow
- Use the Vitis HLS tool for a first project
- Identify the importance of the test bench
- Use directives to improve performance and area and select RTL interfaces
- Identify common coding pitfalls as well as methods for improving code for RTL/hardware
- Perform system-level integration of IP generated by the Vitis HLS tool

Course Outline

Day 1

- **Introduction to High-Level Synthesis**
  Overview of high-level synthesis (HLS), the Vitis HLS tool flow, and the verification advantage. {Lecture}
- **Vitis HLS Tool Flow**
  Explores the basics of high-level synthesis and the Vitis HLS tool. {Lecture, Demo, Lab}
- **Abstract Parallel Programming Model for HLS**
  Describes the structuring of a design at a high level using an abstract parallel programming model. {Lecture}
- **Design Exploration with Directives**
  Explores different optimization techniques that can improve the design performance. {Lecture}
- **Vitis HLS Tool Command Line Interface**
  Describes the Vitis HLS tool flow in command prompt mode. {Lecture, Lab}
- **Introduction to Vitis HLS Design Methodology**
  Introduces the methodology guidelines covered in this course and the HLS Design Methodology steps. {Lecture}
- **Introduction to I/O Interfaces**
  Explains interfaces such as the block-level and port-level protocols abstracted by the Vitis HLS tool from the C design. {Lecture}
- **Block-Level Protocols**
  Explains the different types of block-level protocols abstracted by the Vitis HLS tool. {Lecture, Lab}
- **Port-Level I/O Protocols**
  Describes the port-level interface protocols abstracted by the Vitis HLS tool from the C design. {Lecture, Demo, Lab}
- **AXI Adapter Interface Protocols**
  Explains the different AXI interfaces (such as AXI4-Master, AXI4-Lite (Slave), and AXI4-Stream) supported by the Vitis HLS tool. {Lecture, Demo}
- **Port-Level I/O Protocols: Memory Interfaces**
  Describes the memory interface port-level protocols (such as block RAM and FIFO) abstracted by the Vitis HLS tool from the C design. {Lecture, Lab}
- **Pipeline for Performance: PIPELINE**
  Describes the PIPELINE directive for improving the throughput of a design. {Lecture, Lab}

Day 2

- **Pipeline for Performance: DATAFLOW**
  Describes the DATAFLOW directive for improving the throughput of a design by pipelining the functions to execute as soon as possible. {Lecture, Lab}
- **Optimizing for Throughput**
  Identify the performance limitations caused by arrays in your design. You will also explore optimization techniques to handle arrays for improving performance. {Lecture, Demo, Lab}
- **Optimizing for Latency: Default Behavior**
  Describes the default behavior of the Vitis HLS tool on latency and throughput. {Lecture}

High-Level Synthesis with the Vitis HLS Tool

DSP-HLS (v1.0)

Course Specification

- **Optimizing for Latency: Reducing Latency**
  Describes how to optimize the C design to improve latency. {Lecture}
- **Optimizing for Area and Logic**
  Describes different methods for improving resource utilization and explains how some of the directives have impact on the area utilization. {Lecture, Lab}
- **Migrating to the Vitis HLS Tool**
  Reviews key considerations when moving from the Vivado HLS tool to the Vitis HLS tool. {Lecture}
- **HLS Design Flow – System Integration**
  Describes the traditional RTL flow versus the Vitis HLS tool design flow. {Lecture, Lab}
- **Vitis HLS Tool C++ Libraries: Arbitrary Precision**
  Describes Vitis HLS tool support for the C/C++ languages as well as arbitrary precision data types. {Lecture, Lab}
- **Hardware Modeling**
  Describes hardware modeling with streaming data types and shift register implementation using the ap_shift_reg class. {Lecture}
- **Using Pointers in the Vitis HLS Tool**
  Explains the use of pointers in the design and workarounds for some of the limitations. {Lecture}

Register Today

Morgan Advanced Programmable Systems, Inc. (Morgan A.P.S.) delivers public and private courses in locations throughout the central US region; including Iowa, Illinois, Kansas, Minnesota, Missouri, Nebraska, North Dakota, South Dakota, and Wisconsin.

Visit **morgan-aps.com/training**, for full course schedule and training information.

Morgan
Advanced Programmable Systems

- You must have your tuition payment information available when you enroll. We accept credit cards (Visa, MasterCard, or American Express) as well as purchase orders and Xilinx training credits.

Student Cancellation Policy

- Student cancellations received more than 7 days before the first day of class are entitled to a 100% refund. Refunds will be processed within 14 days.
- Student cancellations received less than 7 days before the first day of class are entitled to a 100% credit toward a future class.
- Student cancellations must be sent here.

Morgan A.P.S. Course Cancellation Policy

- We regret from time-to-time classes will need to be rescheduled or cancelled.
- In the event of cancellation, live on-line training may be offered as a substitute.
- Morgan A.P.S. may cancel a class up to 7 days before the scheduled start date of the class; all students will be entitled to a 100% refund.
- Under no circumstances is Morgan A.P.S. responsible or liable for travel, lodging or other incidental costs. Please be aware of this cancellation policy when making your arrangements.
- For additional information or to schedule a private class contact us here.

Online training with real hardware

- During the Covid-19 period, some companies do not allow their staff to participate in live in-person training.
  - Consequently, Morgan Advanced Programmable Systems, Inc. has set up a training VPN where engineer participants can take classes online using the same computers and devCards used during in-person training.
  - Even better, and upon request, you can use these computers after hours on training days to experiment with labs. This is not possible for in-person training.
  - Additionally, just like in-person training, the laptops and devCards, tools, OS, and licensing are set up in advance.
  - In some ways, live online-training is better than in-person…for example, you can grant the instructor permission to look at your Vivado, PetaLinux terminal, or Vitis for extended periods of time if your lab is not going exactly has planned to a missed step.
  - This is often more comfortable than two engineers crowding around a laptop screen.
  - Taking remote training also allows you to learn some tips and tricks for working remote. Whether your devCard is in the lab down the hall, or across the world via VPN, you can control your Xilinx based device quickly and efficiently.